使用上の注意

【指定可能なポート】

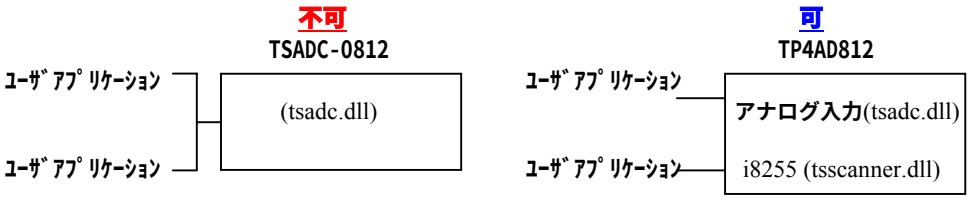
ポートアドレスはベースアドレスからのオフセットを指定してください。実アドレスへの変換はデバイスドライバで行います。
アクセス可能なポートは、デバイスに割り当てられたポートのみです。それ以外のポートにアクセスするとエラーになります。ポート割り当てについては**取扱説明書**を参照してください。

【汎用入出力ドライバ関数】

デバイスのオープン、クローズ、ＩＯ命令関数については同ドライバ付属のドキュメントを参照してください。

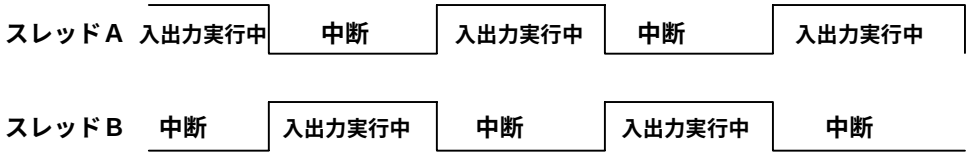
【 デバイスの共有について】

複数のユーザアプリケーション間で同一デバイスを共有することはできません。ただし１つのデバイスが複数のデバイスに分かれていた場合は共有した状態でそれぞれについて制御可能です。



【 入出力周期について】

本ＤＬＬは内部で入出力のためのスレッドを生成します。このスレッドの実行周期はユーザアプリケーションから指定可能です。０ｍｓを指定した場合ＯＳのスケジューリング内で高速に実行します。



また、安定した入出力周期を得るために、 入出力中は他のアプリケーションの実行は控えることをお奨めします。他のアプリケーションが実行中の場合、その分入出力周期は不安定になります。

【参照⇒ＭＳＤＮライブラリ：マルチスレッド プログラム】
【参照⇒ＭＳＤＮライブラリ：Sleep 関数】

●参考として入出力周期０ｍｓでの計測結果を示します。
使用したＰＣ　：Dell Inspiron　4000
ＣＰＵ　インテルモバイル Celeron
内臓キャッシュ　256KB
外部バス周波数　100MHZ

▼ＴＳＳＣＡＮＮＥＲ
３２ポート同時スキャン時――――――――――６００μｓｅｃ周期
１ポートのみスキャン時――――――――――１４μｓｅｃ周期

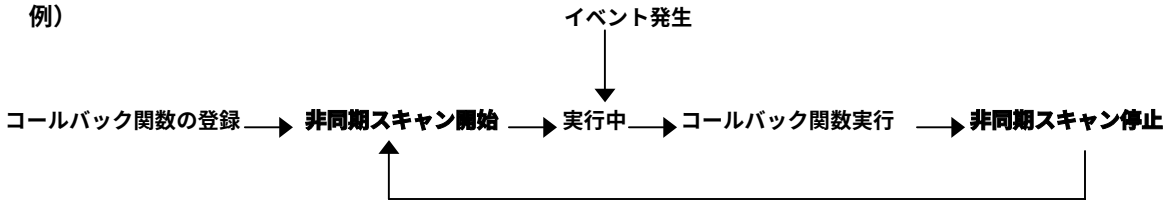
▼ＴＳＡＤＣ
１チャネル実行時の非同期ＡＤ入力時――――７３μｓｅｃ周期

▼ＴＳＣＮＴ
１チャネル実行時の非同期ＣＮＴ入力時―――１００μｓｅｃ周期

【コールバック関数の注意事項】

ＤＬＬが事象検出したとき、この旨をユーザアプリケーションに通知するためアプリケーション内に記述された関数（**コールバック関数**）を呼び出します。
各ＤＬＬのコールバック関数はそれぞれ１つのみしか登録できません。
また呼び出しも一度のみです。再度コールバックしたい場合は非同期ＸＸＸ停止要求を実行してから非同期ＸＸＸ開始要求を再発行してください。

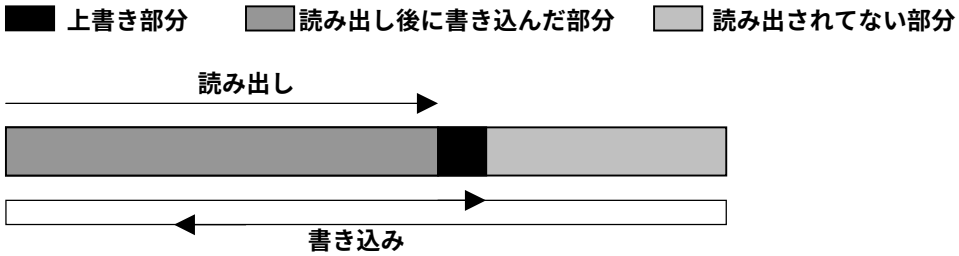
例）



コールバック関数は再入する可能性があります。
このためコールバック関数で共有のスタテック変数等を使用した場合、コールバック関数実行中に再度コールバック関数が呼び出されと、この再入によりスタテック変数の書き換えが起り、再入前の状態に戻ったときに予測できない結果を招くおそれがあります。
従って出来るだけローカル変数を使用するようにしてください。またスタテック変数を使用する場合でも各イベント毎に独立させるなど（配列化）、互いに干渉のないようにプログラミングしてください。

【リングバッファ】

バッファをサイズを超えても書き込み（読み出し）を行えるよう循環制御を施した環状のバッファです。つまりエンドレスのバッファです。
だだし書き込みに読み出しが追従できない場合、その部分は上書きされます。



【Visual Basicでの注意事項】

●コールバック関数は標準モジュール（.bas）に記述してください。
●強制終了、ＥＮＤ、ＳＴＯＰ文について
コールバック関数の登録に AddressOf キーワードを使用するため、Visual Basic の画面で強制終了、ＥＮＤ、ＳＴＯＰ文等の実行はできません。実行するとページ違反が起こります。必ずＥＸＥファイルにしてから実行してください。【参照⇒ＭＳＤＮライブラリ：関数ポインタの制限と危険性】

▼　コールバック関数の登録例

```
--------------------------------フォームモジュール(xxx.frm)-----------------------------------
‘ コールバック関数の登録
ret = TS_DefScanEventHandler(Me.hwnd, AddressOf ScanEventHandler)
If ret < 0 Then
    -----エラー処理-----
End if

---------------------------------標準モジュール(xxx.bas)-----------------------------------
'コールバック関数
Public Function ScanEventHandler( ByVal dhand As Long, _
                                  ByVal EventNo As Long, _
                                  ByVal LParam As Long) As Long
```